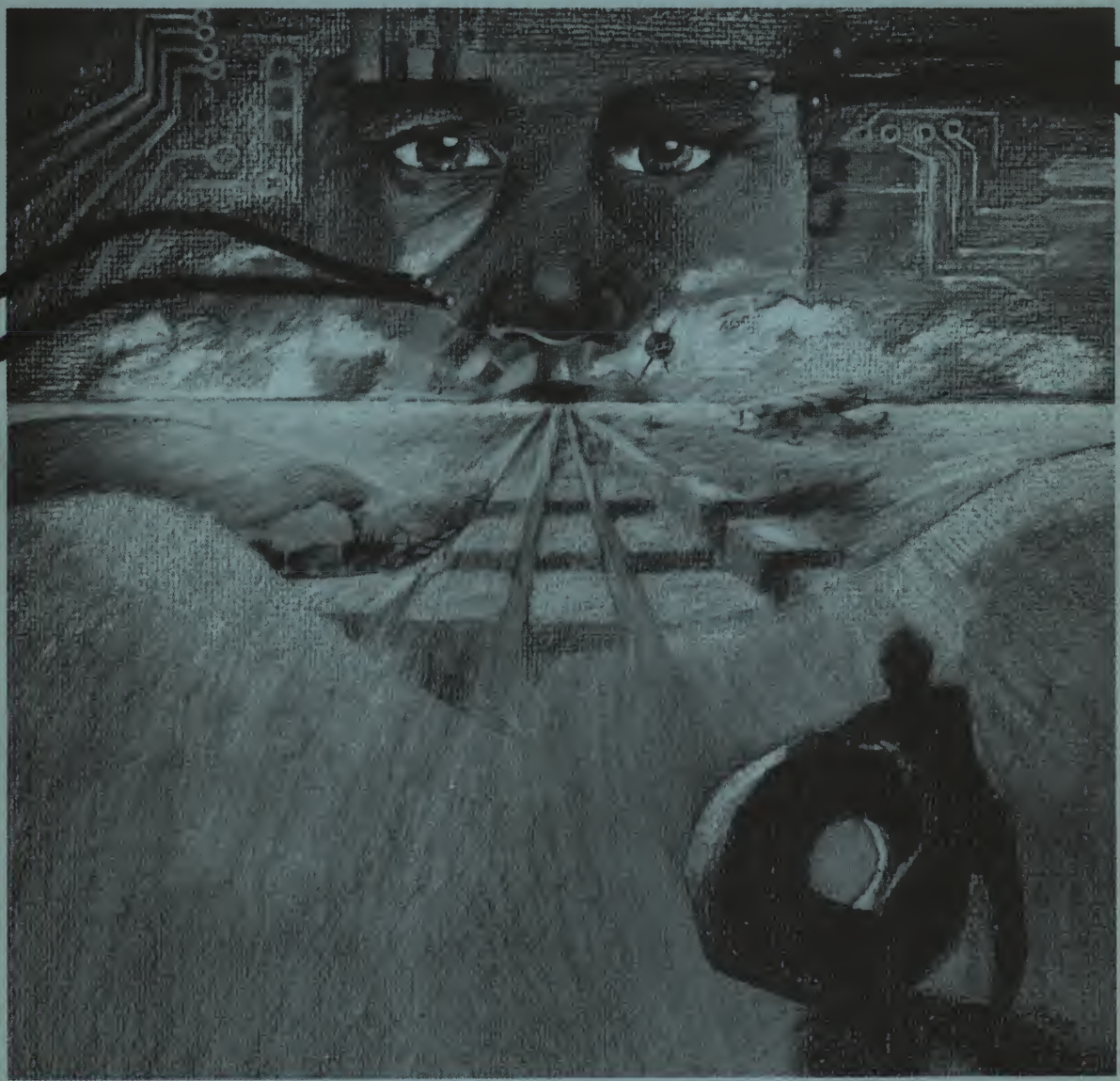


April 1988 nr. 34

Quasar





Laatste redactioneel tje..

Tja, de redactie en Quasar-'bereiding' gaat over in andere handen; Gerard van Rooijen gaat in het vervolg voor de nieuwsbrief zorgen.

Voor de zekerheid herhaal ik hier nog even dat de Quasar een nieuwsbrief is die volledig bestaat uit bijdragen van leden; eigenlijk een uniek iets. Toch kan ik me tegelijkertijd nauwelijks voorstellen dat het op een andere manier wel zou kunnen werken. Als 'samensteller' van zo'n nieuwsbrief heb je altijd net zo weinig tijd als alle leden van de club die kans zien een artikel, een bespreking of een oplossing voor een probleem in te zenden; om elke keer artikelen te schrijven en bovendien een nieuwsbrief samen te stellen is ondoenlijk. Ik hoop dus, voor Sin_QL-air, Quasar, alle leden en vooral voor Gerard, dat jullie net zo actief blijven als in de voorgaande jaren.

Het inzenden van kopij blijft op dezelfde manier gaan; altijd op een cartridge als Quill-file of als listing. Daarbij een verzoekje: als je kopij in Quill aan het typen bent, zet dan de uitlijning (F3-justify) van de tekst op "Left"; dan zet Quill niet op onnoemelijk veel plaatsen dubbele spaties tussen woorden. Na het verzenden van je kopij naar het QOS (Quasar Opmaak Systeem) is er dan minder "Find&Replace" te doen. Ook tabellen zouden geen spaties mogen vertonen, daar hebben we immers de TAB-toets voor!

Iedereen heeft onlangs een acceptgiro ontvangen die eigenlijk voor zichzelf spreekt. De printer van Bob had er een kluif aan, maar voor u betekent het een eenvoudiger manier van betalen. We gaan er van uit dat iedereen die voor eind April nog niet betaald heeft geen lid meer wil zijn.

Mij is het bericht ter ore gekomen dat Komin failliet zou zijn....
Meer hierover misschien in de volgende Quasar.

Groeten van Ron.

NIEUW REDAKTIE-ADRES:

GERARD VAN ROOIJEN
GRUTTOSTRAAT 15
3435 DJ NIEUWEGEIN

BOB VISSER,
Scheepmakerskade
30,
3011 VX Rotterdam.
010 - 414.3554

ADMINISTRATIE,
sekretariaat,
PENNINGMEESTER,
nabestellen oude
nummers.

RON DEN BREEMS,
Kroonstadeef 27,
3067 RT Rotterdam.
010 - 455.1234

REDAKTIE, layout en
samenstelling Quasar,
VOORZITTER.

KEES V.D. WAL,
Kwekerijstraat 22,
2613 VE Delft.
MACHINETAAL,
SUPERBASIC,
PASCAL, HARDWARE-
Quill/Archive/Abacus
vragen.
TUSSEN 20.30 en
23.30 UUR !:
015 - 140.367

MICHEL & WILLEM
SPANJER,
Hortensialaan 11,
3702 VD Zeist.
19.00 - 22.00 uur:
03404 - 20581

REPARATIES,
HARDWARE en
ONDERDELEN
DATABANK-BEHEER

ARD JONKER,
020 - 230.795
MACHINETAAL &
HARDWARE-vragen.

MARC KOOL,
020 - 429.345
C & MACHINETAAL-
vragen.

RUUD
REDDINGIUS,
Ruysdaelstraat 73b,
1071 XB Amsterdam.
020 - 643.544

VERTALEN
ARTIKELEN
BASIC REDAKTIE

Stichting
SIN_QL_AIR,

Rotterdam.
giro:
4597345.

BASIC ARCHIVE

Een computer voor serieus gebruik moet tenminste een tekstverwerker, spreadsheet, database en een programma voor grafieken beschikbaar hebben. Dat vond Sinclair gelukkig ook.

Van de 4 gratis meegeleverde programma's zijn verschillende versies in omloop. Eindelijk is in QLworld en Quasar te lezen wat de verschillen tussen die versies zijn. Neem geen risico en zorg dat u versie 2.3 heeft. De 4 Psion programma's zijn ook voor IBM- klonen beschikbaar onder de naam Pc-four. Psion XChange biedt extra mogelijkheden (Quasar 31: 634-637).

Archive kan meerdere bestanden tegelijk bijwerken en mag dus inderdaad database genoemd worden (Quasar 25: 486-487). Elk bestand bestaat uit een vast aantal velden. Om eventueel toch later een veld toe te voegen moet het hele bestand door een nieuw bestand vervangen worden (Quasar 25: 479-481). Veldnamen die op \$ eindigen zijn strings en mogen uit maximaal 255 tekens bestaan. Overige velden zijn voor numerieke gegevens.

De handleiding beschrijft duidelijk hoe het programma gebruikt moet worden, maar vertelt verder niet hoe ingevoerde gegevens op cartridge / floppy terecht komen. Om te zien hoe een bestand eruit ziet kan het met COPY bestandsnaam to SCR_ zichtbaar gemaakt worden. Na de onzichtbare fileheader heeft Psion er ook nog een etiketje opgeplakt dat uit 20 bytes bestaat: chr\$(0) chr\$(20) vrm1dbf0 chr\$(0) chr\$(0) en 8 bytes die naar bepaalde plaatsen in het bestand verwijzen. De volgorde van de velden van een record is anders dan u met CREATE ingevoerd heeft. Eerst komen alle numerieke velden die per veld 8 bytes innemen. Vervolgens komen alle strings. Elke string wordt voorafgegaan door een teken. De ASCII-waarde van dat teken geeft de lengte van de daaropvolgende string aan. Vandaar dat strings uit niet meer dan 255 tekens mogen bestaan. Na de laatste string wordt het record afgesloten met 2 tekens (chr\$(0)). De

rest van het bestand heeft Archive nodig voor eigen gebruik. De ingevoerde records worden achter elkaar weggeschreven. Bij het sorteren laat Archive het zo en ook het verwijderen leidt niet direct tot een verandering.

Om dit allemaal goed te verwerken heeft Archive dus wel wat ruimte nodig. Een nieuw en nog leeg bestand neemt daarom al 5 sectoren in beslag. Bij per ongeluk gewiste bestanden of na melding van "bad or changed medium" kunnen programma's als Super Media Manager, Cartridge Doctor, Lazarus etc. uitkomst bieden. Bij voor Qdos intacte bestanden kunnen de Psion programma's met meldingen "wrong data type" voor onaangename verrassingen zorgen. Geopende Archive bestanden zijn misschien niet volgens de regels weer gesloten. Met een programma als MDVinspekt_bas (Quasar 26: 513-517; Quasar C11) kan de eerste sector bekeken worden.

Sectornummer noteren. Programma met CTRL+SPATIE afbreken. Intypen: let sector\$(67)="v": put_msec, 1, sectornummer, sector\$. Opnieuw Archive starten en het nog eens proberen. Blijkbaar moet er nog meer veranderd worden, want zelf ben ik er nog niet in geslaagd een bestand op deze wijze te herstellen. Het bestand moet dan zodanig veranderd worden dat het door Archive geïmporteerd kan worden. Een dergelijk exportbestand is veel eenvoudiger van opbouw dan de normale bestanden (User Guide Information: 1-2).

Met een paar regels Superbasic zijn verminkte _doc en def_tmp bestanden weer voor import door Quill gedeeltelijk bruikbaar te maken. Voor Archive is dat helaas minder makkelijk. Gelukkig heeft Chas Dillon in QLUser van januari 1986 een programma gepubliceerd en toegelicht dat dit voor ons doet.

Alle dank aan Kees van der Wal die mij dit artikel toezond. De toelichting is inmiddels zo goed mogelijk in het Nederlands vertaald en bij de software service verkrijgbaar.

Het programma gebruikt in een paar regels Toolkit 2 commando's. Na het starten moet u wel steeds vragen

beantwoorden, maar de ENTER-toets is meestal voldoende. In het begin moet u wel precies de veldnamen in de juiste volgorde intypen. Het is handig als u de creatie van het verminkte bestand in een Archive-programma heeft staan. Als u alle overige procedures verwijdert en alleen een procedure met CREATE ... ENDCREATE overhoudt dan kan het programma hiervan gebruik maken en hoeft u de veldnamen niet in te typen. Heeft u dat niet en weet u het niet meer dan kan het naar scherm kopiëren van het verminkte bestand de oplossing bieden. Ergens aan het einde van het bestand vindt u alle veldnamen in de juiste volgorde. Met dit programma kan natuurlijk geoefend worden op een intact bestand. Als u in strings " gebruikt heeft dan ontstaan er problemen bij het importeren. Hopelijk heeft u dit programma niet nodig, want alles bij elkaar kost het herstellen van een bestand toch aardig wat tijd. Toch maar vaker een kopie maken!

Ik heb in QLWorld gelezen dat als je nu deze kopie in plaats van het origineel voor het verdere werk gebruikt er tijdswinst behaald kan worden, omdat gewiste records dan weggelaten worden en alles op volgorde weggeschreven wordt. Volgens mij is het BACK-UP commando qua werking gelijk aan het Superbasic COPY en wordt de structuur ongewijzigd gelaten. Wilt u toch opruimen dan moet dat als volgt:

```
create "nieuw_dbf" logical "n"
...veldnaam
... endcreat
use "main"
first
while not eof()
...veldnaam=n.veldnaam
... next
endwhile
```

Wie kan het bovenstaande zo aanpassen dat niet alle veldnamen in het programma gebruikt hoeven te worden?

Commentaar is welkom!

Han de Vries Anthony Duyck-
straat 3 8022 AZ Zwolle 038-
532862 ('s avonds)

DE JS AANGEPAST OP EEN 'NIET-QL-PRINTER'

Als je met de QL op een niet QL-printer karakters met een ASCII-kode groter dan 128 probeert uit te printen dan kom je tot de ontdekking dat er van alles uit de printer komt, maar dat dat wel andere karakters zijn dan die je op je scherm zag. Dit komt omdat de QL van sinclair is waar ze het motto "doet iedereen het zo dan doen wij het anders" hoog houden.

Maar in de JS zit een tabel die - als die geactiveerd wordt - elk ascii teken in een ander kan omzetten. De tabel wordt geactiveerd met het kommando "TRA 1,0".

Bij een gewone JS versie zal je niets merken als je dit commando aanroept, dit komt doordat die tabel steeds precies dezelfde ascii code geeft als waar die mee aangeroepen wordt. Het is wel mogelijk om zelf een andere tabel te maken maar die moet steeds na het resetten geladen worden, maar het is ook mogelijk om de tabel in de eeprom te veranderen (als

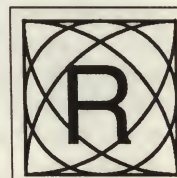
je JS proms of JM proms hebt dan kan je bij Michel Spanjer een JS eeprom setje met handleiding voor het inbouwen kopen). Met de MultiMon kan je de geheugen inhoud bekijken, en op adres b914 begint inderdaad een tabel met alle tekens die de QL kent. De multimon laat gelukkig ook de tekens met een ascii code groter dan 128 zien en zet er niet een punt of blokje neer. Deze tabel moet dus verandert worden, en wel zo dat op de plaats waar nu bijvoorbeeld de ascii code voor de 1 staat, de ascii code waarbij de printer een 1 afdrukt moet komen te staan. Het veranderen gaat in het kort als volgt:

```
-save de inhoud van je 16K JS eeprom
op bandje: sbytes mdv1_js_kode,
(32*1024),(16*1024)
-Laad deze code nu vervolgens in
het geheugen met: a=respr(16*1024)
: lbytes mdv1_js_kode,a : print a
-schrijf de waarde van a even op, en
laad nu de multimon en ga naar adres
a+1556.
```

Hier kan je de tabel veranderen. Hier na verlaat je de multimon en save je de nieuwe code met sbytes mdv1_js2_kode, a, (16*1024) en deze code kan je met een eeprom programmer op eeprom zetten die als hij in de QL zit, na aanroep van tra 1,0 inderdaad de goede tekens laat uitprinten.

Ook kan je natuurlijk de foutmeldingen vertalen of je naam bij het opstarten op het scherm laten zetten door de tekst "1983 Sinclair Research Ltd" te veranderen. Deze staat iets verder dan de tabel in de eeprom.

Roelof Crevecoeur



In het januarinummer van QL-WORLD zagen we de advertentie van een nieuw toetsenbord voor de QL; de eveneens in dit blad vermelde test ervan gaf ons zoveel vertrouwen dat we besloten het te bestellen, samen met de vermelde stofkap; wel prijzig, nl. Hfl. 403.20 incl. BTW, ingeklaard en franko thuis.

Te gek als je bedenkt dat je al een goede complete gebruikte QL kunt kopen voor rond de Hfl. 200.-.

Door een fors ongeluk sneuvelde een van onze QL's, maar wonder boven wonder was heel de inhoud nog in orde; alleen de kast had het begeven. De beloofde levertijd van 28 dagen was waarschijnlijk bedoeld voor binnen Engeland zelf, want bij ons duurde het 48 dagen. We maakten het pakket open en troffen aan: Het keyboard, de solide lange spiraalkabel met plug (nu kun je het bord, makkelijker dan eerst, een andere positie geven, b.v. op schoot in een gemakkelijke stoel), de stofhoes, de interface, en het deksel dat in de plaats van het oude toetsenbord komt en waarop de monitor geplaatst kan worden. Verder een velletje ombouwbegeleid-

SCHÖN PC KEYBOARD

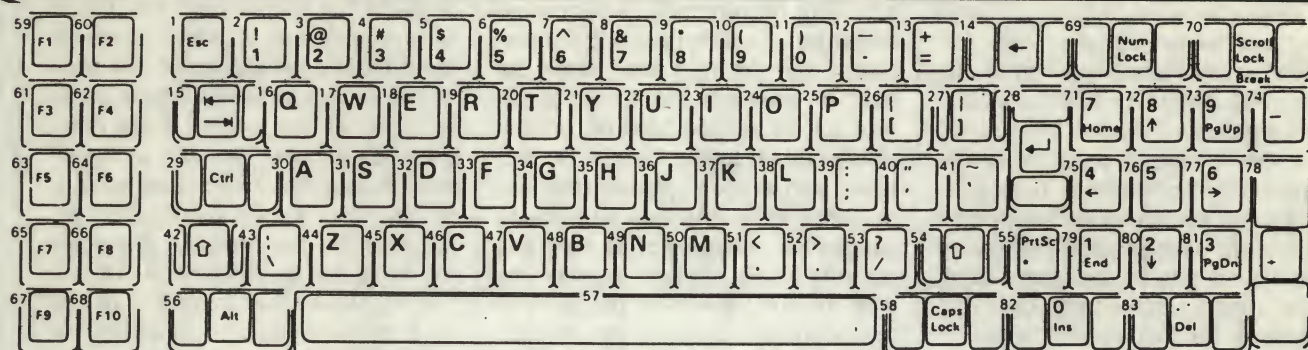
ing; Het benodigd gereedschap is een kruiskopschroevendraaier en een gewone. Het ombouwen duurt maar kort en kan door iedereen zelf gedaan worden; bij ons was dit iets gecompliceerder omdat de bodem van onze verongelukte QL na het lijmen van de stukken niet meer zo mooi in model bleek en omdat een paar steuntjes hierin verloren waren gegaan.

Hoe werkt het en wat is het verschil? Degelijk en prettig in de aanslag, ramelt niet en werkt gedempt; de kast is stevig en licht en de toetsen hellen iets meer dan de originele QL. Er zitten rubber nokjes onder deze kast en ze blijft stevig staan, zonder te bewegen of te schuiven; geen uitklapbare steunen die het board echt flink schuin zetten, en soms wat wankeler maken. Caps Lock is voorzien van een rood ledje dat brandt bij in bedrijf; dit is ook het geval als het extra nummeriek toetsenbord is ingeschakeld en deze uitbreiding is enorm voor diegenen die veel cijferwerk in moe-

ten voeren en al of niet gewend zijn aan de posities van de toetsen van gewone tel- en rekenmachines. Wissen naar links of rechts gaat fijner; je hoeft er maar een toets voor in te drukken. F 1 t/m F 5 zijn ongewijzigd; de uitbreiding F 6 t/m F 10 in hetzelfde blok, ook tomatrood, is erg mooi maar of dit ook inwendig meer mogelijkheden geeft weten we nog niet; er is geen "gebruiksaanwijzing" bij over nieuwe mogelijkheden, maar de wijzigingen die we al ontdekt hebben werken alleen maar prettiger. De kleur van de kast is fris beige met een zacht vleugje pastelgeelgroen erin; de hoofdtoetsen zacht lichtgrijs en de funktietoetsen iets donkerder; het deksel over de computer heen is zwart als de QL bodem zelf en alle aansluitingen van voorheen zijn gebleven. Andere nieuwe (grote) toetsen zijn Sys Req, Break, Pause, Return en 0.

C.H.M. Biemans, Veghel.

HET AANSLUITEN VAN EEN ASCII TOETSENBOORD OP DE QL.



Note: Nomenclature is on both top and front face of keybutton as shown. The number to the upper left designates the button position. ④

Vele QL-gebruikers zullen het QL-toetsenbord als hinderlijk ervaren. Zeker als je er nog het een en ander aan je QL hebt hangen is het verveelend dat je geen los toetsenbord hebt. Mede hierom heb ik een "toetsenbord interface" gemaakt waarvan hieronder enige uitleg. Er bestaat (nog) geen printplaatje voor dus tenzij je dat zelf kunt maken zal je alles met draadjes aan elkaar moeten binden.

HET TOETSENBOORD:

Alhoewel iedereen waarschijnlijk verwacht dat er uit een ASCII-toetsenbord een seriele ascii-kode komt is dit toch niet het geval. Iedere toets heeft namelijk twee eigen codes (een voor het indrukken en een voor het loslaten). De twee shift toetsen hebben dus ook ieder twee eigen codes. Stel dat je bijvoorbeeld de combinatie Control-Shift-Q indrukt, dan geeft het toetsenbord eerst een code voor het indrukken van de control (29), vervolgens een code voor het indrukken van de shift (42 of 54), dan voor het indrukken van de Q (16) en vervolgens de codes voor het loslaten in de volgorde waarin je de toetsen loslaat. In totaal zijn er dus zes codes verzonden. De codes voor het loslaten vind je door bij de codes voor het indrukken 128 op te tellen. Het is namelijk zo dat elke code bestaat uit 8 bits en dat het MSB (Most Significant Bit) aangeeft of het een toets is die wordt ingedrukt of wordt losgelaten. Op het eerste gezicht denk je: waar is dat nu weer goed voor, nog zo'n code terwijl er toch al een ascii-kode bestaat,

maar met dit systeem is het mogelijk om dezelfde IBM machines in verschillende landen te gebruiken, waar dan alleen een andere versie van het bios (Basic I/O System) op moet draaien. Maar we hebben het hier niet over IBM's, maar over een toetsenbord. Als er een code uit het toetsenbord komt dan wordt er tegelijkertijd een trein van clock-pulsjes meegegeven. Deze trein is niet 8 bits lang, maar 9 en ik denk dat het eerste bitje er voor dient om aan te geven dat er een code aankomt. Ik heb dat overigens niet gebruikt. We weten nu genoeg om het interface te bouwen. Op afbeelding 1. staan de 'indruk codes' vermeld bij de toetsen en op afbeelding 2. staat hoe de 5-polige din plug moet worden aangesloten.

HET INTERFACE:

Ik heb het geheel van TTL bouwstenen gemaakt, op een onderdeel na waarvan geen TTL versie van bestaat, maar hierover verderop meer. Het data-sigitaal uit het toetsenbord wordt ingelezen in een schuif-register (bijv. de 74LS164) door het clock signaal uit het toetsenbord aan te sluiten op de clock van het schuif-register en de data uitgang in te lezen op ingang A. Als de voeding en de clear goed zijn aan gesloten komt er nu op de uitgang van het schuif-register de code uit het toetsenbord in binaire vorm te staan. Omdat er tijdens het inlezen van het schuifregister allemaal onzin op de uitgang van dit schuifregister staat wordt de code die op de uitgang van het schuif-register staat na het schuiven ingelezen in een latch (bijv.

de 74LS373) waarvan de ingang disabled is tijdens het schuiven. Dit wordt bereikt door de enable ingang van deze latch aan te sluiten op de uitgang van een one-shot (bijv. de 74LS123) die getriggerd wordt op het clock signaal uit het toetsenbord. De one-shot geeft dus een puls tijdens het inlezen van de code in het schuifregister. Aan het einde van de puls uit de one-shot wordt dus de code uit het schuifregister in de latch ingelezen. Deze code bestaat uit een 7-bits code plus een bitje (MSB) dat aangeeft of de toets wordt ingedrukt of losgelaten. De 7-bits code wordt aangesloten op de 7 laagste adreslijnen van een 8 K eeprom (2764). Dit kan eventueel ook een 4 K eeprom zijn maar deze is vaak duurder. Deze eeprom heeft 8 uitgangen en kan dus - mits hij goed geprogrammeerd is - gebruikt worden om 8 elektronische schakelaartjes aan te sluiten. Maar dat is veel te weinig, want in de QL zitten meer dan 50 toetsen, dus als je voor elke toets een schakelaartje neemt, dan zou je 7 eeproms nodig hebben, iets wat alleen maar node-loos duur wordt. Ten eerste kan de eeprom een aantal tabellen bevatten die om de beurt in verschillende latches worden ingelezen, die op hun beurt weer de schakelaartjes besturen. Maar dan moeten er nog altijd 7 latches met meer dan vijftig schakelaartjes aangesloten worden. Dit is eigenlijk helemaal niet nodig als je bedenkt dat er nooit meer dan 1 toets tegelijk ingedrukt wordt. (Uitzonderde shift, ctrl, alt en de cursor toetsen) Je kunt namelijk 1 centrale draad nemen die de twee connectors van de

HET AANSLUITEN VAN EEN ASCII TOETSENBORD OP DE QL.

QL verbind en waar de 9 resp. 11 aansluitingen met schakelaartje mee verbonden worden (zie afbeelding 3). Als er nu een bepaalde toets geselecteerd wordt, dan worden twee schakelaartjes in doorlaat gezet, waardoor er verbinding komt tussen twee punten van de verschillende connectoren. Anders gezegd komt het er op neer dat het ene schakelaartje de rij en het andere schakelaartje de kolom selecteert. De shift, control, alt en de cursor toetsen krijgen ieder een eigen schakelaartje. We hebben het aantal benodigde schakelaartjes nu terug gebracht tot 23 (8 om de rij te selecteren, 8 om de kolom te selecteren en 7 voor de control, shift, alt en de cursor toetsen).

Maar het zou natuurlijk mooi zijn als toetsen als Del, Page Down, Page Up, Insert/Overwrite, Scroll Lock, Home, End, F6 t/m F10, * en + het ook zouden doen. De QL heeft deze functies namelijk ook allemaal wel, maar dan met een combinatie van twee toetsen. Je kan om dit op te lossen nog een tweede serie schakelaartjes te nemen die parallel staan aan de eerste serie, maar

die met vertraging worden aangestuurd. Bij zo'n combinatie van toetsen is de tweede toets die je indrukt nooit een ctrl, alt of shift dus we hoeven voor de 'tweede serie' alleen maar 8 kolom schakelaartjes en 8 rij schakelaartjes te nemen. In totaal hebben we nu vijf tabellen in de eprom nodig. (1 om de rij schakelaartjes, 1 voor de kolom schakelaartjes, 1 voor de ctrl, shift, alt en cursor toetsen, 1 voor de vertraagde rij schakelaartjes

en 1 voor de vertraagde kolom schakelaartjes). Er is dan ook nog het probleem van de 'num-lock' toets die heen en weer schakelt tussen de cijfers en de cursor toetsen op het kleine rechter toetsenbordje. Dit probleem kan worden opgelost door aan de 7-bits kode nog een bitje toe te voegen dat aangeeft of de num-lock actief is of niet. In feite worden alle tabellen dus twee maal precies hetzelfde, met uitzondering van de toet-

ledje op het toetsenbord synchroon te laten lopen met de aanturing van de tabel moet deze flipflop bij het aanzetten in een bepaalde stand gezet worden met een R-C tijd. Deze verschillende tabellen worden achter elkaar geselecteerd door een tellertje te laten lopen (bijv. de 74LS93) en daarvan de drie hoogste uitgangen aan te sluiten op de adreslijnen 8 t/m 11. Op de adressen 0 t/m 6 komt dan de 7-bits kode te staan en adreslijn 7 wordt gebruikt om aan te geven of de num-lock aan of uit is. Bovendien worden alle vier de uitgangen van dit tellertje aangesloten op een decoder (bijv. de 74LS154). Met de uitgangen van deze decoder kunnen nu de verschillende latches "enabled" of aangestuurd worden. Ik heb steeds eerst de tabel laten selecteren en pas een stand verder van de klok de desbetreffende latch enabled, zodat de eprom alle tijd heeft om z'n zaakjes op orde te krijgen. Als de teller zijn standen doorloopt gebeurt er dus het volgende:

Tellerstand:

b0000

De tabel op adres 0 wordt geadresseerd. Deze bevat de codes voor de shift,ctrl,alt,num-lock en cursor-toetsen.

b0001

De latch voor deze schakelaartjes wordt enabled.

b0010

De tabel op adres h100 wordt geadresseerd. Deze bevat de codes voor de rij-schakelaartjes.

b0011

De latch voor de rij-schakelaartjes wordt enabled.

b0010

De tabel op adres h200 wordt geadresseerd. Deze bevat de codes voor de kolom-schakelaartjes.

b0011

De latch voor de kolom-schakelaartjes wordt enabled.

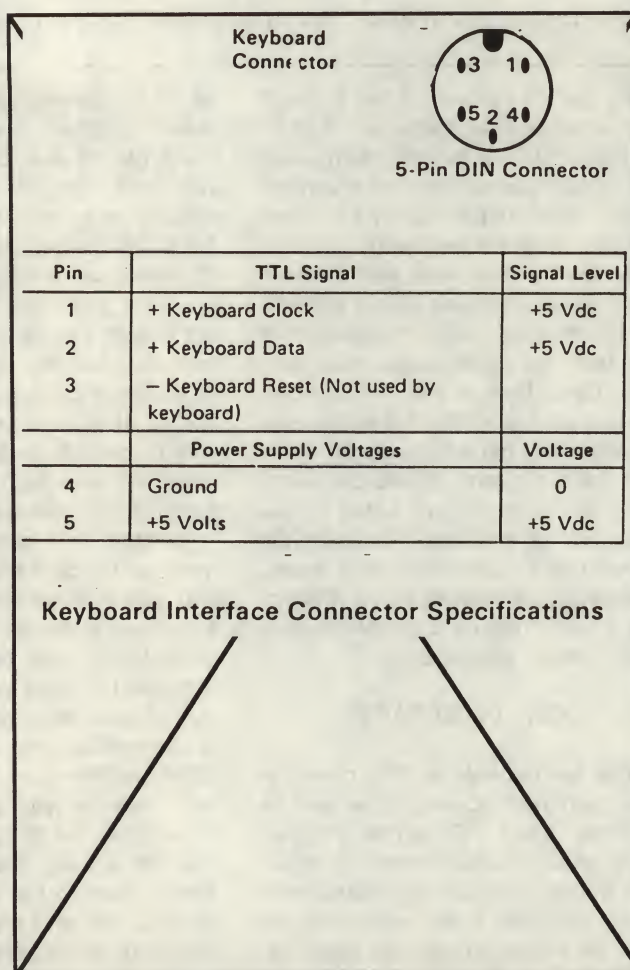
b 0 1 0 0

- b 1 0 0 1

Gebeurt niets.

b1010

De tabel op adres h500



sen op het rechter toetsenbordje, in de eprom gezet. Elke keer dat de num-lock toets wordt ingedrukt komt er net als bij alle andere toetsen een kode uit, die wordt gedetecteerd in de tabel bij de shift, ctrl, alt en de cursor toetsen en dus ook de num-lock. Via een flip-flop in de toggle mode wordt dan de zevende adreslijn aangestuurd die dus bepaalt of de tabel met de cursor toetsen of de tabel met de cijfers wordt geselecteerd. Om het

HET AANSLUITEN VAN EEN ASCII TOETSENBOORD OP DE QL.

wordt geadresseerd. Deze bevat de codes voor de vertraagde rij schakelaartjes.

b1011

De latch voor de vertraagde rij schakelaartjes wordt geselecteerd. b1100

De tabel op adres h600 wordt geadresseerd. Deze bevat de codes voor de vertraagde kolom schakelaartjes.

b1101

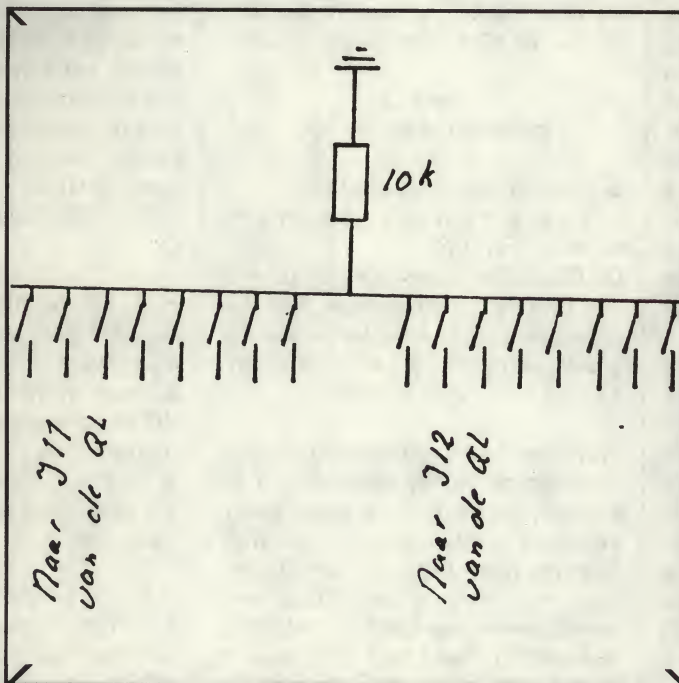
De latch voor de vertraagde kolomschakelaartjes wordt geselecteerd.

Zoals je ziet heb ik de afstand tussen het aansturen van de directe schakelaartjes en de vertraagde schakelaartjes zo groot mogelijk genomen, omdat ik hoopte dat de tijd die hiertussen zit groot genoeg was voor de vertraaging. Helaas was dit niet het geval en daarom heb ik de output control van de latches die de vertraagde schakelaartjes aansturen geregeld met een one-shot

die getriggerd wordt op de flank van het signaal uit de decoder. (steeds wordt er een one shot gemaakt vlak voor de waarden uit de tabellen h500 en h600 in de latches worden gelezen zodat die waarden nog niet meteen op de uitgang van de desbetreffende latch komen.) Het was niet mogelijk om het klok-sig-naal van het tellertje met een lagere frequentie te nemen omdat de totale verwerkingstijd dan de te

langzaam werd. Het ascii-toetsenbord heeft namelijk ook een auto-repeat met codes die vrij snel achter elkaar komen, en als het klok-sig-naal dan te traag is kan het voorkomen dat de laatste code (het loslaten) niet gedetecteerd wordt en de toets dus voor de QL 'ingedrukt blijft'.

Nu moet de eeprom nog geprogrammeerd worden. De tabel op adres 0 is vrij eenvoudig, want alleen op de adressen waar een shift, ctrl, alt, num-lock en de cursor toetsen worden gebruikt moet iets komen te staan. Op alle andere adressen in deze tabel komen nullen. Op afbeelding 1. zie je bijvoorbeeld dat ctrl een code 29 heeft, dus op adres 29 komt een 1 (als het ctrl schakelaartje wordt aangestuurd met uitgang 1 van de desbetreffende latch). Op de adressen 42 en 54 (de shift toetsen) komt dan bijvoorbeeld een 2 te staan en op adres 56 (alt) een 4, adres 69 (num-lock) een 8 etc. Dus altijd een macht van 2, anders dan zouden er meerdere schakelaartjes tegelijk aangezet worden. Op adres h100 komt dan de tabel voor de rijen. Op adres 16 bijv. (q) komt 64=26 want de q staat op de zesde rij in de QL-matrix. En in de tabel op adres h200 (voor de kolom schakelaartjes) komt op adres 16 een 8 omdat de Q in de derde kolom staat.



Bij toetsen als Scroll Lock (adres 70) moet zowel in de tabel die op adres 0 begint een code staan (1 voor de ctrl), als in de tabellen voor de vertraagde rij en kolom besturing (1 voor de rij en 32 voor de kolom. F5 is immers de nulde rij (20=1) en de vijfde kolom (25=32). Natuurlijk moet het

tellertje ook weer gereset worden. Ik heb dat gedaan door stand 15 te detecteren met een drie nand porten en deze geïnverteerd met een vierde nand poort. Dit signaal gaat via een flipflop naar de reset ingang van het tellertje. Het tellertje blijft zodoende niet gereset omdat het door de reset stand 0 in neemt en zodoende z'n eigen reset opheft. Overigens kan de stand waarop de teller gereset moet worden natuurlijk ook uit de decoder worden afgeleid, maar je moet hem toch inverteren dus die nand porten heb je toch nodig. De teller wordt gestart door de neergaande flank van de one-shot te detecteren die wordt gegenereerd tijdens het 'inschuiven' in het schuifregister. Met de vijf latches die uit de eeprom worden ingelezen worden nu CMOS schakelaartjes (4066) aangestuurd. En de aansluitingen van deze schakelaartjes worden nu op de connectoren J11 en J12 van de QL aangesloten. (Dit zijn de connectoren waar die papieren

lintkabels in vast zitten geklemd). Toen ik zover was werkte alle toetsen wel maar ze begonnen direct met repeteren. Dus je kreeg voor elke letter kreeg je er drie of vier op je scherm. Ik begrijp nog steeds niet waarom dit gebeurt, maar het is op te lossen door een weerstandje van 10K te leggen tussen de aarde en 'centrale draad', die de kolomschakelaartjes met de rij-schakelaartjes verbind. Zie ook afbeelding 3.

Toen ik zoever was werkte vrijwel alles behalve dat het mis ging als ik een combinatie van ctrl-shift of ctrl-alt of alt-shift gebruikte want dan ging het bij sommige toets mis, dat wil zeggen dat er twee letters op het scherm kwamen. Dit komt doordat het I.B.M. toetsenbord er blijkbaar niet tegen kan als je combinaties van meer dan twee toetsen indrukt. Maar ook dit is eenvoudig te verhelpen door in serie met de shift, ctrl en alt schakelaartjes van het IBM toetsenbord een diode te zetten. En jawel, de QL staat nu ergens achteraan op mijn buro onder een hoop troep terwijl ik dit getikt heb, kortom het werkt!

DATA-COMMUNICATIE MET EEN NULMODEM.

Het overzetten van files van de Sinclair QL (onder het Q-DOS) naar een machine onder MS-DOS (IBM-cloon) zonder gebruikmaking van communicatieprogramma's, maar met behulp van een 0-modem kabel.

Zo'n twee en een half jaar geleden kocht ik een Sinclair QL met de bedoeling om er veel tekstverwerking (QUILL) mee te doen en om er bestanden (m.b.v. ARCHIVE) mee op te bouwen. Inmiddels heb ik een boekmanuscript voor de helft af (ruim 250 k) en een octrooien-bestand t.b.v. dit boek van zo'n 70 k. Ondanks dat ik er een floppy-drive aan heb hangen, begon ik tegen teveel beperkingen aan te lopen. Het interne geheugen van de standaard 128 k schreeuwde om uitbreiding en verder was het maken van backups naar een andere floppy alleen via de tijdrovende weg van de microdrives mogelijk. Ook kreeg ik steeds meer behoefte aan een wat uitgebreidere tekstverwerker met meer mogelijkheden van tekens (QL-printer driver 10 tekens tegen WP 2 x 26 vertaal- mogelijkheden). Ook de mogelijkheid van het werken met grote woordenboeken (thesauri) en spellingchecker waren aantrekkelijk. Om een lang verhaal kort te maken: er kwam dus een Cloon in huis: 640 k intern en een 30 Meg. harde schijf. Dus nu nog even de QUILL-docs overzetten en we kunnen het hele Word-Perfect geweld op de hoofdstukken loslaten, Jawel:

Ik zal U het verslag van de vele geïnvesteerde uurtjes / telefoontjes en diepe zuchten en vloeken van mijzelf en twee leden van het gewaardeerd HCC-bestuur afd. Nijmegen (Rudolf en Rene) besparen. (o.a. omdat in de KOMIN-informatie hierover nogal wat fouten zaten). Hieronder volgt de methode die wij hebben gebruikt en die goede resultaten heeft opgeleverd.

Allereerst bespreken we hoe je files overzet van QL naar "IBM", daarna de omgekeerde weg.

1. Van QL naar "IBM"

Zorg eerst voor een 0-modem verbinding tussen QL en IBM. Hiervoor hebben we de standaard kabel gebruikt voor een seriele QL-printer.

Deze kabel heeft al de zeer moeilijk verkrijgbare stekker die in de seriele poort van de QL moet en aan de andere kant een standaard seriele plug. (N.B.) deze laatste moet nog wel omgesoldeerd worden volgens het hierbijgevoegde schema).

Omdat bovengenoemde kabel maar een meter lang is kan een tussenkabel (verlengsnoer) erg handig zijn. (zie schema)

Begin met het ontvangstklaar maken van de IBM:

- a. opbooten
- b. na prompt type: mode com1: 4800, n, 8, 2 (ENTER)
- c. er verschijnt dan : com1: 4800, n, 8, 2, -

Nu heb je verschillende mogelijkheden: je kunt de file naar verschillende opslagmedia overbrengen: naar drive c (harde schijf) of naar drive a of drive b bijv. naar drive b type: Copy aux b:filename (ENTER) Naar de harde schijf: Copy aux:filename (ENTER) Na enter verschijnt een knipperende cursor : de IBM is ontvangst- klaar.

Stap 2 Instellen van de QL

- a. Type: baud 4800 (ENTER)
- b. Type: Copy_n mdv1_filename_lis to ser1z (ENTER)

De QL begint nu de print-file (in ASCII) over te sturen naar de IBM. De tijdsduur is o.a. afhankelijk van de grootte van de file (bij 4800 baud zo'n 480 c.s. of +/- 2 sec. per kb.).

Wanneer het oversturen gelukt is, verschijnt op de IBM de melding: 1 file copied. en de prompt komt terug. Wanneer er iets mis gaat (bijv. wanneer het geen lis-file betreft of wanneer met verschillende baudrates wordt gewerkt dan komt op de IBM de foutmelding: Read fault error reading device AUX abort, retry, ignore kies dan voor abort en check juiste procedure (zie hierboven).

Ad. a. Baud 9600 bleek problemen te geven met de handshaking die bij baud 4800 niet voorkomt. Omdat in dit geval de snelheid van overdracht niet zo belangrijk is en feitelijk nauwelijks merkbaar, gebruiken we baud

4800.

Ad. b. Voordat we een file overzetten moeten we zorgen dat deze in de juiste vorm staat. In ASCII en liefst met zo min mogelijk tekstverwerkers-stuurcodes (bold/underlined etc.). Hiervoor maken we eerst een printer-driver die "zo leeg mogelijk" is. Daarna maken we vanuit QUILL een lis_file aan. (Document met de extension _lis).

Dit gaat als volgt:

F3 print (enter) current of type de naam van de te printen file (enter) whole to printer. Op de plaats van printer typen we het medium waar-naar de printfile moet worden weggeschreven bijv. mdv1_ Op het opslagmedium komt dan de lis file te staan. Het is raadzaam om even te controleren hoe deze file eruit ziet door deze naar het beeldscherm te kopiëren m.b.v. het commando: copy_n drivename (bijv. mdv1_) to con_

Bij gebruik van de floppy-drive lukte het overzetten niet in de default-mode (na opstart mdv1_), maar moest eerst met FSET de micro-drive_mode default worden gemaakt. Dan de floppy met FDK of FLP aansturen: Copy_n fdk1_filename_lis to ser1z (ENTER).

2. Van MS/DOS-machine (cloon) naar QL

Het over te sturen document (bijv. een W.P.-file) moet eerst in de goede vorm staan.

a. Maak er een DOS-file van door in WP het gewenste document te laden. (current doc).

b. Daarna CTRL F5 (tekst in/uit) keuze 1 = save. Geef hierbij pad en/of filename aan

N.B. Zet bij voorkeur deze file in de ROOT DIR., omdat we nl. terugmoeten naar de root (prompt) om het DOS_commando te kunnen geven: Mode com1:4800,n,8,2 (ENTER)

Na de prompt typen we:

Copy drivename:filename.extension aux (n.b. pas een return geven wanneer de QL in ontvangst-mode staat!).

Wil je eerst zien wat je gaat overzenden copieer dan naar het scherm d.m.v. het commando: copy aux con

DATA-COMMUNICATIE MET EEN NULMODEM.

Om de QL ontvangstklaar te maken typen we:

baud 4800 (ENTER) copy_n ser1z to device-name_filename_exp (ENTER)
De QL maakt dan eerst een dir-naam aan en staat dan ontvangst- klaar.

Als er op de MS/DOS-machine iets mis gaat kan de foutmelding: "Write fault error writing device aux" optreden. Om te voorkomen dat de QL in ontvangst-mode blijft staan wachten, omdat er bijv. iets mis is gegaan: Veroorzaakt dan bewust een fout om de cursor terug te krijgen door bijv. nog iets met een afwijkende baud-rate te versturen.

Als laatste advies: Heb enig geduld bij het oversturen. Het gedrag (o.a. tijdsduur) van de ontvanger is on-

voorspelbaar!

Schema verlengkabel:

Alles direct doorverbinden volgens schema o- modem kabel:

ser-plug QL

Printerzijde

1	zwart	>	7
2	groen	>	3
3	wit	>	2
4	blauw	>	20
5	rood	>	5
6	oranje	>	6

N.B. Zie zonodig ook de informatie in de QL-User guide onder CONCEPTS:
p. 13-14 : Communications RS-232-C
p. 16-17 : devices
(KOMIN in Eindhoven en Ard Jonker van de Sin_QL_air gebruikersgroep)

zijn eveneens met bovenstaande problematiek bekend).

Henk Cloosterman
(i.s.m. Rudolf Ruiters en Rene Janson).

Eventuele reacties/vragen naar: HCC
afd. Nijmegen Postbus 6601 6503
GC Nijmegen.

SPELLINGS-CONTROLE, AUTO'S & VIDEO.

Weet je zonder enige aarzeling te vertellen wat de juiste schrijfwijze bij de volgende voorbeelden is ?

onmiddellijk	-	onmiddellijk
handvaten	-	handvatten
copieën	-	copiën
enveloppen	-	enveloppen
commissie	-	comissie

Zelfs indien je precies zou weten waar in het bovenstaande de fouten zitten, is een spellingschecker bijna onmisbaar om serieus te kunnen tekstverwerken. Of denk je soms dat je nooit typefouten maakt, nooit eens verkeerd aanslaat ?

Quill is standaard niet uitgerust met een mogelijkheid voor spellingscontrole. Anders gezegd, Quill is niet in staat om ingevoerde woorden te toetsen aan een, in het geheugen aanwezig, Engels of Nederlands woordenboek.

QSpell

Het oudste, en lange tijd enige, programma voor spellingscontrole is QSpell. Het programma doet ongeveer wat je mag verwachten, maar kent enkele serieuze beperkingen:

- De omvang van het referentie-

woordenboek is aan een maximum gebonden. Normaal gesproken kan een spellingschecker in het gebruik 'bijleren', d.w.z. nieuwe woorden toevoegen. QSpell is na enige tijd 'vol'.

- Qspell kan niet multitasken, terwijl het bovendien vrij veel tijd kost om het programma te laden. Het is dus niet simpel om vanuit Quill een document snel even op fouten te controleren.

- Indien Qspell een fout in een document ontdekt, kan die fout niet op eenvoudige wijze in het origineel gecorrigeerd worden. Sinds kort adverteert QJump met het programma QTYP.

Voordien was SPELLBOUND het enige andere programma voor spellingscontrole op de QL.

Spellbound als Maserati

Over het algemeen wordt er positief over Spellbound geoordeeld. In de QL World van juli 87 werd het programma zelfs met een soort halleluja-geroep verwelkomd. Het slot van die bespreking is te mooi om hier niet te citeren: "The Sector Software sister product, Taskmaster, was described

in a recent review as the Rolls Royce of the multi-taskers. In the same vein Spellbound must be described similarly but with some of the attributes of a Maserati. If you try to cross a Maserati with a Rolls Royce, you can easily become spellbound." Als je dit laatste letterlijk neemt, lijkt het er inderdaad een beetje op dat de bespreker betoverd geraakt is. Op deze conclusie valt naar mijn oordeel namelijk wel wat af te dingen. Om te beginnen is Taskmaster geen Rolls Royce. Met flink wat goede wil kun je er misschien nog een Mercedes of een Rover in zien. Meer niet. Spellbound moet echter in de goedkopere prijsklasse gezocht worden. Misschien is een hele eenvoudige Citroën een betere vergelijking: sympathiek eigenwijs maar met structurele fouten in het ontwerp. (Overigens heeft Citroën enige jaren geleden een tijdlang een model met Maserati-motor geproduceerd. Dit type, de SM, werd echter een flop). Nu zijn er best een paar redenen om Spellbound te prijzen. Het aardigste van Spellbound is wel dat het een 'on-line' spellingschecker is. Het principe is simpel: alles wat je typt wordt onmiddellijk gecontroleerd en kan eventueel ter

SPELLINGS-CONTROLE, AUTO'S & VIDEO.

plekke gecorrigeerd worden. De meeste andere programma's voor spellingscontrole, inclusief QSpell, kunnen slechts na afronding van het hele document aan het controleren slaan. Prettig is ook nog dat het woordenboek van Spellbound onbeperkt kan groeien. Spellbound kent dus geen van de nadelen die aan QSpell kleven. Een laatste pluspunt van Spellbound is dat het gebruik niet tot Quill beperkt is: ook andere programma's zoals de Editor of Archive (zie Quasar 32) kunnen eenvoudig met Spellbound gecombineerd worden.

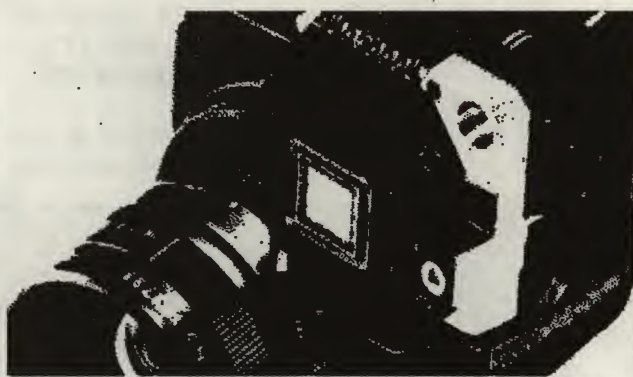
Spellbound als lelijke eend

De eerste (voor mij fatale) constructiefout van Spellbound is dat het programma zichzelf bij het minste of geringste uitschakelt. Zodra er een cursorpijltje gebruikt wordt is Spellbound verdwenen en moet het eerst weer met CTRL + O geactiveerd worden. Dit is om compleet gek van te worden. Wie Quill of de Editor niet alleen maar gebruikt om tekst over te typen, maar vooral gebruikt om zelf teksten samen te stellen, zal regelmatig een paar woordjes of regels willen terugspringen om nog iets aan de formulering te veranderen. (Ook terwijl ik bijvoorbeeld dit stukje schrijf, gebruik ik met grote regelmaat de cursorpijltjes om hier en daar wat bij te schaven.) Misschien is zo'n flexibele correctiemogelijkheid zelfs het essentiële verschil tussen een tekstverwerker en een schrijfmachine. En juist deze mogelijkheid wordt door dat vervelende automatische uitschakelen gefrustreerd. Een tweede ernstige tekortkoming van Spellbound is het onvermogen om gelijktijdig en probleemloos met Keydefine, het macroprogramma, gebruikt te worden. Zonder het onvolprezen Keydefine zijn Quill en de Editor alleen met zeer omslachtige commando's te bedienen en niet half zo bruikbaar.

Filebound

Filebound is geen zelfstandig programma, maar een accessoire (mooi woord voor spellingscontrole!) bij Spellbound. Aangevuld met Filebound kun je Spellbound als 'klassieke'

spellingschecker gebruiken en hele documents controleren. Handig is vooral dat je op die manier op simpele wijze het woordenboek sterk kunt uitbreiden door Spellbound grote hoeveelheden nieuwe woorden (uit oude documents) te laten 'leren'. Tegen een geringe meerprijs wordt Filebound bij Spellbound geleverd. Daarnaast werd de listing van dit door Chas Dillon geschreven programma gepubliceerd in de QL World van november



1987. Omdat ik nieuwsgierig & gek genoeg was, heb ik het programma overgetypt en uitgeprobeerd. Het resultaat viel niet mee: het oude QSpell werkt in ieder geval beter dan de tergend trage combinatie van Spellbound en Filebound. Een standaardgrap in allerlei manuals is het advies om bij bewerkingen die de computer veel tijd kosten 'zelf maar even thee te gaan zetten'. Als Filebound aan het controleren slaat behoort de bereiding van een volledige maaltijd echter tot de mogelijkheden. Om snelheid te winnen heb ik Filebound met behulp van Turbocharge gecompileerd. Dat hielp, maar verrichtte nog steeds geen wonderen. Om een idee te geven: een tekst met een bescheiden omvang van 1000 woorden (ongeveer dit stukje tot op dit punt) wordt bij de basic-versie in 25 minuten gecontroleerd. De gecompileerde versie is met zo'n 8 minuten ongeveer drie keer zo snel. (Nou ja, snel) In ieder geval hebben mijn associaties meer met trek-schuiten dan met Maserati's te maken. Voor de bezitters van Spellbound die interesse hebben, heb ik de basic- en turbo-versie van Filebound naar Kees gestuurd om in de cartridge-service opgenomen te wor-

den.

Tekstverwerken met de QL ?

Het bovenstaande was helaas niet erg positief. Mijn conclusie is dan ook dat er, in ieder geval op dit moment, geen echt goede spellingschecker voor de QL bestaat. Helemaal afgezien van de bekende klachten over toetsenbord, microdrives en Quill, betekent dit dat de QL als tekstverwerker het moet afleggen tegen computers als de Amiga, Atari ST of de eindeloze stroom van MS-DOS machines. Voor deze computers kan namelijk met het modieuze, maar goed doordachte Word Perfect gewerkt worden, terwijl voor die 'Charlie Chaplin-

machientjes' bovendien nog een handvol andere goede programma's voor spellingscontrole & tekstverwerking bestaat. Misschien ben ik te somber en blijken de programma's QTYPE of TEXT 87 van hoge kwaliteit te zijn. Bovendien bewijzen de Editor en Turbocharge dat er nog steeds serieus aan de perfectieering van software voor de QL gewerkt wordt. Toch doet alles me een beetje aan de ontwikkelingen op videogebied denken. Om een aantal redenen werd het systeem dat eigenlijk het beste was, Video 2000 van Philips, uiteindelijk de grote verliezer. Het in wezen tamelijk beperkte VHS-systeem heeft voor een wereldstandaard gezorgd. Wordt dan toch nog MS-DOS het VHS voor computers ?

Dick Verroen

Het Gebruik van pointers in Metacomco Pascal.

Eigenlijk heb ik mij nooit bezig gehouden met pointers in pascal, omdat ik altijd dacht dat het iets ingewikkelds was, zonder dat er een echt nuttige toepassing voor bestond. Nu begrijp ik dat de echte pascal liefhebber meteen aan het grommen of iets dergelijks slaat, want de mogelijkheid tot het gebruik van pointers in pascal is toch "het neusje van de zalm" als het om programmeren gaat. Het zal allemaal best wel waar zijn, maar ik heb er toch te weinig kaas van gegeten om dit echt te kunnen beamen (of tegen te spreken).

Nu moest ik een poosje geleden veel in de programmeertaal PL/M-86 programmeren en hierin kun je echt niet om pointers heen. Een pointer in deze taal is niets anders dan een variabele waarin een geheugenadres opgeslagen kan worden, bijvoorbeeld de locatie van een variabele. Parameters van procedures of functies kunnen in PL/M nooit gestructureerde variabelen zijn, maar wel een pointer naar de gewenste structuur. Door een '@' voor de (gestructureerde) variabele te zetten, b.v. met @klant, wordt niet de variabele klant benaderd, maar het adres waar de variabele klant staat opgeslagen. Maar ik had het over pascal en hierin zijn pointers toch iets heel anders, of niet??

Ik programmeer nu in Turbo pascal op een MS/DOS-systeem en daarin zit de mogelijkheid calls te doen naar het operating system. Dit had ik b.v. nodig om vanuit turbopascal een directory op het beeldscherm te kunnen toveren. Ik dacht natuurlijk meteen aan de QTRAP-procedure in Metacomco-pascal voor een zelfde toepassing. In de manual staat een voor-

beeld hoe je QTRAP kunt gebruiken om de INKEY\$-functie uit BASIC na te bootsen, maar andere toepassingen liepen bij mij al snel vast. Het probleem dat ik had bij veel QDOS-calls was, dat een pointer naar een buffer in register A0 of A1 gezet moet worden. Bij mijn weten bestaat er geen functie in Metacomco-pascal die de geheugenlocatie terug geeft van een variabele (zoals LOC(var) in Computer One-pascal, blz. 655 Quasar). Ik moest natuurlijk meteen aan de pointer uit PL/M denken om zo het fysieke adres van een variabele te kunnen bemachtigen. En ja hoor, pointers in pascal bleken niet iets 'engs' te zijn en leken verdacht veel op de pointers die ik gewend was uit PL/M. De syntax is iets anders en er mag iets minder (maar daar is het pascal voor). Toch blijkt een pointer variabele in pascal niets anders te zijn dan een geheugenlocatie. Nu was het voor mij dus eindelijk mogelijk een pointer naar een buffer mee te geven. Het volgende pascal-programma, dat een directory van een device laat zien, maakt hier gebruik van. Hierin worden dus pointer-variabelen gebruikt om zowel input- als output-buffers mee te geven aan QDOS-calls.

De procedure dir opent een directory-file. Om een directory te openen moet het te openen device (lengte WORD gevolgd door device-naam) aangewezen worden door A0. De procedure dir moet dus aangeroepen worden met als parameter een PACKED ARRAY OF CHAR waarin de eerste twee posities worden opengelaten om het lengte-WORD in te zetten. Dus b.v. dir('***mdv1_'); In een directory-file staan alle fileheaders (64 bytes) opgeslagen en in zo'n fileheader staat

o.a. de naam van de file. Vanaf het 14e byte begint dit met een lengte WORD (2 bytes) gevolgd door maximaal 36 karakters (bytes) die de naam voorstelt. De procedure dir positioneert de file-pointer aan het begin van de volgende fileheader en leest vervolgens 38 bytes (2+36) in. De buffer waarin deze 38 bytes komen te staan wordt aangewezen door de pointer Regs2.A1@. Dit wordt herhaald tot dat het einde van de file word bereikt (Reg3.D0 = ERR.EF = -10). Ik weet eigenlijk niet waarom ik hier niet gewoon Regs1 kan meegeven in plaats van een nieuwe variabele Regs3, maar anders werkt het niet (Heeft iemand hier misschien wel een verklaring voor?)

Misschien dat iemand iets heeft aan dit artikelje of wellicht zijn er op- of aanmerkingen. Ik hoor deze graag. Ik weet niet of er veel mensen in Metacomco-pascal werken, maar het is misschien wel handig een library van algemeen toepasbare procedures aan te leggen.

Reactie graag naar
Joost Mulder
Nieuwe Keizersgracht 6III
1018 DR AMSTERDAM
020-265988.

Bij voorbaat dank



PASCAL

```
PROGRAM Directory(input,output);
```

```
TYPE MediumType = PACKED ARRAY[1..7] OF CHAR;
  { [1..2] = length-WORD,
    [3..7] = medium-name }
VAR CharNr: INTEGER;
    medium: MediumType;
```

```
PROCEDURE dir(medium: MediumType);
```

```
CONST CurrentJob = -1;
    Timeout = -1;
    DirectoryType = 4;
    MaxFileName = 38; {2 BYTES length-WORD,
        36 BYTES file-name }
    IOpen = 1;
    IClose = 2;
    IOFstrg = 3;
    FSposab = 66;
    NoError = 0;
    FileNameStartPosition = 14;
    FileHeaderLength = 64;
```

```
TYPE IOType = (open,close);
```

```
    FileNameType = PACKED ARRAY[1..MaxFilename] OF
    CHAR;
```

```
    RegPack1= RECORD
        D0: INTEGER;
        D1: INTEGER;
        D2: INTEGER;
        D3: INTEGER;
        A0: @MediumType;
        A1: INTEGER;
        A2: INTEGER;
        A3: INTEGER
    END;
```

```
    RegPack2= RECORD
        D0: INTEGER;
        D1: INTEGER;
        D2: INTEGER;
        D3: INTEGER;
        A0: INTEGER;
        A1: @FileNameType;
        A2: INTEGER;
        A3: INTEGER
    END;
```

```
VAR Regs1 : RegPack1;
    Regs2 : RegPack2;
    FileName : FileNameType;
    FileNr,
    DirectoryChannelID,
    FileNameLength,
    CharNr : INTEGER;
    EndOfFile : BOOLEAN;
```

```
PROCEDURE Directory(WhatToDo: IOType); {** Opens
```

```
or closes a directory file **}
```

```
BEGIN
CASE WhatToDo OF
    open : Regs1.D0:=IOpen;
    close: Regs1.D0:=IClose
END;

Regs1.D1:=CurrentJob;
Regs1.D3:=DirectoryType;
NEW(Regs1.A0); { Create var. }
Regs1.A0@:=medium; { Vul var. }
Regs1.A0@[1]:=CHR(0); { Set char- }
Regs1.A0@[2]:=CHR(5); { count to 05 }
QTRAP(2,Regs1,Regs2); { Call IO.???? }
DirectoryChannelID:=Regs2.A0; { Save ChannelID }
DISPOSE(Regs1.A0) { Kill var. }
END;
```

```
PROCEDURE FilePointerAtFileName(nr: INTEGER);
{Sets filepointer at filename(nr)}
BEGIN
    Regs2.D0:=FSposab;
    Regs2.D1:=FileNameStartPosition +
        (FileHeaderLength * nr);
    Regs2.D3:=Timeout;
    Regs2.A0:=DirectoryChannelID;
    QTRAP(3,Regs2,Regs2);
END;
```

```
PROCEDURE GetFileName(VAR FileName: FileName-
Type);
{Reads FileName}
VAR regs3: RegPack2;
BEGIN
    Regs2.D0:=IOFstrg;
    Regs2.D2:=MaxFilename;
    Regs2.D3:=Timeout;
    Regs2.A0:=DirectoryChannelID;
    NEW(Regs2.A1);
    QTRAP(3,Regs2,Regs3);
    FileName:=Regs2.A1@;
    EndOfFile:=(Regs3.D0 <> NoError);
    DISPOSE(Regs2.A1)
END;
```

```
BEGIN {dir}
    Directory(open); { Open Directory-file }
    FileNr:=0; EndOfFile:=FALSE; { Initialise }
    WRITE('Directory of volume ');
    FOR CharNr:=3 TO 7 DO WRITE(medium[CharNr]);
    WRITELN; WRITELN('-----');
    REPEAT
        FilePointerAtFileName(FileNr);
        GetFileName(FileName);
        FileNameLength:= ORD(FileName[1])*16
            + ORD(FileName[2]);
        IF FileNameLength>0
        THEN BEGIN
```


PASCAL

```
FOR CharNr:=1 TO FileNameLength DO
WRITE(FileName[CharNr+2]);
Writeln;
END;
FileNr:=SUCC(FileNr);
UNTIL EndOfFile; { No other files found }
Directory(close) { Close Directory-file }
END;
```

```
BEGIN {PROGRAM Directory}
WRITE('I want to see a list of all files on volume: ');
FOR CharNr:=3 TO 7 DO
  READ(medium[CharNr]);
  Writeln; dir(medium)
END.
```



BASIC

LISTING.

```
10 REMark Van M.J. Wheeler uit QL World Dec.'87.
20 REMark In december 1987 voor QL(JS_ROM) aange-
past
30 REMark door J.W.P. Bartholet te MOORVELD.
40 REMark
50 REMark --- Character Write Routine ---
60 REMark De QL_basis_ en alternatieve karakterset
70 REMark in alle maten, standen en QL_kleuren.
80 REMark
90 REMark Z a$,b,c,d,x,y
100 REMark a$ = het af te beelden karakter.
110 REMark b = afmeting van een blokje in pixels.
120 REMark c = elke hoek in radialen t.o.v.
130 REMark horizontaal naar rechts. (b.v. 2*PI=360'
140 REMark PI=180';PI/2=90';PI/4=45';(1/9)*PI/4=5')
150 REMark d = de kleur van het karakter.....
160 REMark ALS voor d 'n kleurnummer, groter dan 7
170 REMark gekozen wordt, krijgt elke 'pixel' van
180 REMark elke te printen karakter 'n random kleur
190 REMark tussen 2 en 7; ongeacht karaktergrootte.
200 REMark x = x_co-ord. linksboven van karakter.
210 REMark y = y_co-ord. linksboven van karakter.
220 REMark *****
230 REMark
500 DEFine PROCedure Z (a$,b,c,d,x,y)
510 LET f=b*COS(c)
520 LET g=2*b*SIN(c)
530 FOR loop0=0 TO 8
540 LET h=PEEK(((44453)+(CODE(a$)-32)*9)+ loop0)
550 IF CODE(a$)>127 THEN h=PEEK(((45329)+
```

```
(CODE(a$)-127)*9)+loop0)
560 FOR loop1=0 TO 7
570 IF d>7 THEN e=RND(2 TO 7):ELSE e=d
580 IF h>=2^(7-(loop1+1)) THEN
590 BLOCK b,b,x+loop1*f-loop0*g,y+((loop1*g/
4)+loop0*f),e
600 LET h=h-2^(6-loop1)
610 END IF
620 END FOR loop1
630 END FOR loop0
640 END DEFine Z
650 REMark *****
660 REMark D E M O
670 REMark *****
1000 WINDOW 480,256,16,0:PAPER 0:CLSE0:CLSE2
1010 kop1
1020 golf
1030 kop2
1040 kop3
1050 sinclair
1060 QL
1070 REMark *****
1080 REMark
1100 DEFine PROCedure kop1
1110 sp1$='Speciale Procedure'
1120 x0=246
1130 y0=70
1140 FOR telsp=1 TO 18
1150 hoek=RAD(telsp*(120/(LEN(sp1$)+1))-(60+120/
(LEN(sp1$)+1)))
1160 x=x0+190*SIN(hoek)
1170 y=y0-50*COS(hoek)
1180 Z sp1$(telsp),2,hoek,4,x,y
1190 END FOR telsp
1200 INK 4:ARC 32,78 TO 106,79,-1.3*PI/6
1210 END DEFine kop1
1220 DEFine PROCedure golf
1230 sp1$='om karakters te printen'
```



```

1240 x=40:y0=100
1250 d=2
1260 FOR telsp1=1 TO 23
1270 x=x+16
1280 y=y0
1290 IF (LEN(sp1$)-(23-telsp1)) MOD
2 = 0 THEN y=y0-4
1300 Z sp1$(telsp1),1.5,0,d,x,y
1310 d=d+1
1320 IF d=8 THEN d=2
1330 END FOR telsp1
1340 kl=2:x=14:y=53:hk=PI/8
1350 FOR tel2=1 TO 2
1360 FOR tel=1 TO 3
1370 FOR tel1=1 TO 2
1380 ink kl:ARC x,y TO (x+18),y,hk
1390 x=x+18:hk=hk-kl:kl=kl+2
1400 IF kl>7 THEN kl=2
1410 END FOR tel1
1420 END FOR tel
1430 hk=-hk:x=14:kl=kl+2
1440 END FOR tel2
1450 END DEFine golf
1460 DEFine PROCedure kop2
1470 sp2$=' voor de.... '
1480 x0=246
1490 y0=130
1500 FOR telsp2=1 TO 16
1510 hoek=-RAD(telsp2*(180/
(LEN(sp2$)+1))-(270+180/
(LEN(sp2$)+1)))
1520 x=x0-190*SIN(hoek)
1530 y=y0-65*COS(hoek)
1540 Z sp2$(telsp2),4,-
(hoek+135),4,x,y
1550 END FOR telsp2
1560 END DEFine kop2
1570 DEFine PROCedure kop3
1580 sp3$='? ? ? ? ? '
1590 x=215
1600 y=130
1610 d=2
1620 FOR telsp3=1 TO 11
1630 Z sp3$(telsp3),8,0,d,x,y
1640 d=d+1
1650 IF d>7 THEN d=2
1660 END FOR telsp3
1670 PAUSE 150:CLS
1680 END DEFine kop3
1690 DEFine PROCedure sinclair
1700 si$='SINCLAIR'
1710 x=15:y=10
1720 FOR telsi=1 TO 8
1730 Z si$(telsi),4,0,2,x,y
1740 x=x+5:y=y+30
1750 END FOR telsi
1760 END DEFine sinclair
1770 DEFine PROCedure QL
1780 ql$='QL'
1790 x=80:y=0
1800 FOR telql=1 TO 2
1810 Z ql$(telql),36,0,8,x,y
1820 x=x+195
1830 END FOR telql
1840 END DEFine QL
1850 REMark VEEL PLEZIER

```

SPECIALE KARAKTER-PRINT- ROUTINE.

Deze routine geeft je de mogelijkheid om de gehele QL-karakterset inclusief de alternatieve set te printen in formaten tot beeldschermhoogte toe. Daarby kun je de groep karakters tot ongeveer beeldscherm-hoogte printen in ALLE richtingen en standen; cirkelvormig, gegolfd, vertikaal etc. noem maar op. Zelfs achterstevoren of ondersteboven kan. Bij deze routine wordt met behulp van de karakterinfo, opgeslagen in de ROM en het BLOCK-commando elke pixel van elk karakter apart op het scherm geprint. Hierdoor is het in principe dus ook mogelijk om elke pixel apart in te kleuren.

Deze routine is afkomstig uit de QL-WORLD van Dec.'87 en ontworpen door M.J.Wheeler. Ik heb ze licht aangepast waardoor o.a. de pixels in random gekleurd kunnen worden.

Eventueel dient er 'n ander adres ingevuld te worden. De listing hieronder geldt voor het JS-ROM. Een ander ROM, dan een ander adres invullen en klaar.

Een eigen karakterset kan ook gebruikt worden.

Een nadeel is dat de routine erg traag is en een aparte printvorm kwa programmeren bewerkelijk kan zijn. Echter ideaal om een resultaat van deze routine als beeldscherm in een Desktop-Publishing-programma in te laden.

John Bartholet uit MOORVELD.



!!!

**John Bartholet uit MOOR-
VELD.**



VRAAG & AANBOD.....

Te koop aangeboden:

Schön-toetsenbord. Desgewenst monteer ik het ook. De prijs is fl. 90,-
 • Tandata-modem (bestaande uit Q-Connect, Q-Mod en Q-Call), software en documentatie: fl. 200,-
 • Turbocharge compiler v. 1.12 + tool-kit, met originele documentatie fl. 45,-

• Super Media Manager v. 1.12, met originele documentatie fl. 45,-
 • Taskmaster v. 2.35, met originele documentatie fl. 45,-
 • Keydefine, met originele doc. fl. 25,-
 Bovenstaande prijzen zijn exclusief eventuele verzendkosten.

Dick Verroen 020-465214

Te koop aangeboden:

Sinclair QL - JS., 640 Kb extern. Monitor Philips M80, amber, printer Brother HR5, boeken en tijdschriften, plm. 50 cartridges, héél veel software!
 Prijs f700,-

Hans, 080 - 566069, 's-avonds.

Te koop aangeboden:

2x QL (waarvan één met defect); compleet met manuals, kabels, etc. 1x trumpcard, 1x ice-rom met muis, artice, choice, icicle. 1x PLCM interface met 3.5" diskdrive. 1x MP 256K internal mem. exp.
 Software: special editor v1.6 en v1.8 met manual, turbo-quill, cosmos, locksmith & 4matter, copycat en 128k boot, pro astronomer, cartridge doctor, home finance, qspell, ql-utilities, spellbound, 20 disks en een doos met slot, 30 cartridges en twee doosjes, advanced user guide, maintenance manual, spare ic's.
 Alles in één koop: f1500,-

**C.R. Worthington
 Gelderse Kade 80A
 1012 BL Amsterdam
 prive: 020-200577
 werk: 020-5982242**

Te koop aangeboden:

Sinclair QL 512k (intern) Version JS met reserve membraam f475.
 Ancona (Philips CM 8533) kleuren-monitor geheel instelbaar f 500.
 CST diskinterface + CST (NEC) DS/DD diskdrives 3.5" dus 2x720k. inclusief eigen ingebouwde voeding f700.
 Sinclair QL (Seikosha SP 1000 QL) printer met alle QL karakters en grafisch (bv. PSION's EASEL) f475.
 Polyphemus (Elektuur) modem PTT goedgekeurd + Miracle Systems Modaptor 300/300, 1200/75 en 1200/1200 baud met kabels en software f150.
 Transform box inclusief 18 cartridges (2x beschikbaar) f 60. Originele software: Cartridge Doctor (Talent) f20. Steve Davis Snooker (CDS Software) f10. QL Reversie (Sinclair) f10. QL RAM Disk + Utilities f20. Copycat (Compware) f15. Boeken 7 stuks voor f40. Tijdschriften QL-World 22x f30. Your Spectrum / Your Sinclair 21x f25. Joystick (Suzo Competition Pro) f15. PC10 en PC15 (TDK) datacassettes f1. p.st.

H.Blom 070 - 238636

Te koop aangeboden:

M.C.C. Pascal (rom)
 vr.pr. f225,-

010-4147347

Te koop aangeboden:

Printer TPX 1000 met ser/par interface incl. 3 rollen therm. papier: f250,-. Monitor Ancona high res. f150,-. R.C. QJump TWEE (ORG.) met handleiding f25,-. R.C. ICE met handleiding f25,-. 15 cartridges met prog. f30,-. BOEKEN: QL Superbasic van Ian Jones f20,-. QL Technical Guide f15,-. Sinclair QL gebruiker-shandboek f15,-. Handleidingen: superforth, home finance, tascopy, tasprint, supercharge, super-media-manager, pcb-designer, desktop publisher, turbo compiler, the editor, monql, etc. f60,-. Alles in één koop f450,- (vaste prijs.) **013-550617** na 18.00. Gust de Beer, Tilburg.

GEVRAAGD:

Diskdrive interface voor 2 drives 360 Kb. **Bellen: wo. t.e.m. za.**
04130 - 63224 (Henk)
 na 18.00 uur op woe-do-vr.

WIE KAN MIJ VERDER HELPEN?

Ik ben in het bezit van twee stuks BASF 6106 diskdrives 5.25'. Nu wil ik deze gaan aansluiten op mijn QL-JS met CST1.13, maar dit lukt niet. Wie heeft hiermee ervaring of kan suggesties geven om dit probleem op te lossen?

Hardy Klink.
05490-25737 na 18.30

Gevraagd:

QDOS-companion (eventueel ruilen) Source voor alle mogelijke soorten programma's voor Metacomco 'C' en 'BCPL' en voor Computer One Pascal en Forth.

a.u.b. even schrijven naar:
Peter Venema,
Westereems 1,
9642 KP Veendam.

Gevraagd:

Wie kan mij aan een 3.5' disk-interface helpen? Liefst van CST. Of het volledige schema hiervan?

Tom van der Woude
Purmerend
02990-32470

Te koop aangeboden:

QL-JS.
 Seikosha 500 AS printer.
 20 cartridges.
 3 'The Sinclair QL'-serie boeken.
 Instructieboeken+ service manual.
 Prijs f500,-

N. Schaap, 013 - 551395

And now for something completely different...

door een QL-junkie.

(het zijn zijn eigen woorden...®)

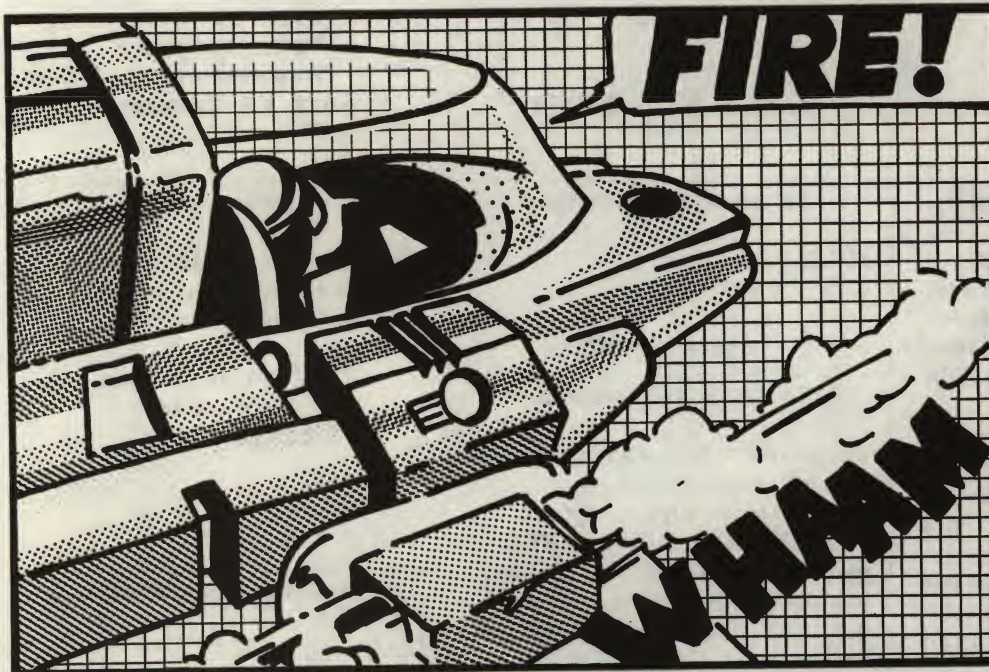
Velen van ons zullen zich de spelen 'Knight Lore' en 'Alien 8' op de gouwe ouwe Spectrum nog wel herinneren. Het kon dus niet uitblijven of er zou zo iets dergelijks voor ons aller QL verschijnen. CHI-SOFT kwam nu met een spel-programma dat eindelijk eens laat zien wat het verschil is tussen een 8-bitter en een 'volwassen' 16-bitter. Ik had een redelijke verwachting van het spel toen ik het eindelijk in handen kreeg, maar het bleek toch nog veel mooier te zijn. Vooral omdat het gehele spel 192 3D-schermen bevat en toch maar zo'n 60k groot/klein is. Het spel 'ALIEN HI-JACK' speelt zich af in een soort ruimteschip, dat door de buitenaardse wezens is gekaapt. Afijn dat verhaal kent u natuurlijk allemaal. De bedoeling is dat men 3 codes verzamelt en hierdoor je macht weer over-

neemt. Het is een echte grafische avonture in de manier dat je echt allerlei middelen moet bedenken om ook maar wat verder te willen komen. Bijvoorbeeld je moet in een paar schermpjes, met veel pijn, moeite en levens (in het begin), twee krukjes ophalen om ergens anders op een paar tonnen te klimmen en daar een sleutel te pakken die je nodig hebt om een anders gesloten deur te openen. Kortom leuk maar erg moeilijk (misschien het leukste). Het spel bevat, voor zover ik gekomen ben, leuke gevarieerde schermen, en uiteraard de meest vreemde monsters, die je het liefst naar een andere wereld zouden helpen. Sommige van die beesten kun je met n schot doden, maar andere hebben er zelfs tot 8 stuks nodig, of je kunt ze zelfs helemaal niet dood schieten. Het speelveld bestaat uit de kamer waarin je je bevindt, en daaronder een console waarop allerlei informatie staat over de behaalde punten, je resterende ener-

gie, kogels, de verzamelde voorwerpen en alle gebruikerstoetsen. Een groot pluspunt vind ik dat je er levens bij kunt krijgen, door een soort smurf op te pakken die een leven voorstelt. Ook vind ik het heerlijk dat er een pause mogelijkheid opzit, voor de zenuwen. Jammer vind ik wel dat je geen situaties kunt wegsaven, want het is een hoop werk om ver te komen. Nu ben ik geen echte spelletjes liefhebber, want die gaan mij al snel tegenstaan en vervelen, maar dit is een spel wat toch nog enig denkwerk vereist, en voor menigeen toch avonden lang leuk zal blijven.

ALIEN HI-JACK werd mij 'beschikbaar' gesteld door £16,- over te maken aan CHI-SOFT te LONDON

Tom van der
Woude



HOOFDSTUK ZEVEN cursus MACHINETAAL: -LOOPS- door ard jonker

Er bestaan twee verschillende soorten loops. In basic kennen we die als de FOR...NEXT loop en de REPEAT UNTIL loop. In machinetaal kunnen we een zelfde onderscheid maken. De For...NEXT loop zullen we het eerst behandelen. Hiervoor is een speciale instructie bedacht. Eerst een voorbeeld van hoe het kan, maar het eigenlijk met een 68000 niet moet.

```
START  MOVE.L Loop_index,D0
```

```
Loop
*   doe iets
    ADD.L  £$-1,D0
    CMP.L  0,D0
    BNE   Loop
```

In basic kunnen we dit als volgt 'vertalen'

```
10 D0 = Loop_index
20 REMark doe iets
30 D0=D0-1
40 IF D0 < 0 THEN GO TO 20
```

```
Netter zou zijn
10 D0 = loop_index
20 REPEAT Loop
30 REMark doe iets
40 D0=D0-1:IF D0 = 0 THEN EXIT
```

```
Loop
50 END REPEAT Loop
```

Hier kunnen we duidelijk zien waar de loop zich bevindt. Zeker als we meer dan 1 loop in elkaar hebben, leid het gebruik van GO TO tot zogenaamde spaghetti-programma's. De nettere versie in machine taal is

```
START  MOVE.L  Loop_counter-1,D0
```

```
loop
*   doe iets
    DBF  D0,loop
*   ga verder
```

We hebben kortere en duidelijker code. De instructie DBF -in feite de instructie DBcc, met voor cc (conditie-code) een 'F' ingevuld- is zeer krachtig. DBF betekent decrement and branch until false. Oftewel: verlaag register en spring naar label indien niet waar. De conditie 'niet

waar' kan vervangen worden door bijvoorbeeld 'negatief', waarbij de instructie DBN ontstaat. De sprong naar genoemd label -loop- wordt dus uitgevoerd als aan de ene kant de teller de teller nog geen -1 is, en aan de andere kant de conditie nog niet vervuld is. De conditie wordt getest door het betreffende bit in het conditiecode register te testen. Dit gebeurt automatisch.

De instructie test de inhoud van het opgegeven register. Is de waarde van dit register niet kleiner(!) dan 0, en bovendien niet de voorwaarde van de instructie voldaan (in het geval van False nooit!) dan springt hij naar het aangegeven label. Is de waarde van het register kleiner dan 0, dan loopt het programma verder bij "*" ga verder'. Hiermee hebben we ook de gemene klem te pakken: als je 10 keer iets wilt doen, dan zal pas de elfde keer de inhoud van het register op -1 staan. Als we dus de bovenstaande 'doe iets' opdracht 10 keer willen uitvoeren, lijken we de loop_counter op te moeten 10 stellen. Dan zou -omdat de teller pas aan het einde getest wordt- de loop 11 keer worden doorlopen. Daarom trekken we bij het laden van de loop_counter er alvast 1 af. De loop begint dan met de counter op 9. Na 1 loop wordt deze verlaagd tot 8, en na tien keer de loop te hebben doorlopen staat de teller op -1 en breekt de loop af.

De andere loop waarover we hebben gesproken is de loop die uitsluitend afhankelijk is van een bepaalde test. Om tests uit te voeren zijn een aantal mnemonics bedacht. Zo zijn er CMP -compare- en BRcc. Branch on condition. Branch is een sprong. We kunnen afhankelijk van een bepaalde voorkomende gelegenheid, conditie, een bepaalde sprong ondernemen.

Zo kunnen we bijvoorbeeld de gebruiker vragen een toets in te drukken. Zolang er geen toets wordt ingedrukt springen we terug en vragen we opnieuw om een toets in te drukken. Pas als er een bepaalde toets wordt ingedrukt gaan we verder met het

programma. Het vergelijken van de toets die is ingedrukt met een door ons gewenste toets gaat bijvoorbeeld met CMP. Als we de ingedrukte toets in ascii waarde in register D0 krijgen, en we willen het programma alleen verder laten gaan als er een 'j' of een 'n' is ingetoetst, kunnen we het volgende stukje programma inbouwen:

```
get_key
```

```
.....(stuk programma dat de ingedrukte toets in ascii
```

```
.....waarde in D0 zet)
```

```
CMPI.B 106,106 'j' = ascii 106
```

```
BEQ cont_prog
```

```
CMPI.B 110,D0 'n' = ascii 110
```

```
BEQ cont_prog
```

```
bra get_key
```

```
cont_prog
```

We zien dat er twee maal getest moet worden om te controleren of een 'j' dan wel een 'n' is ingetypt. Is geen van beide toetsen ingedrukt, dan springen we opnieuw terug naar het stukje programma dat een toets in leest; get_key.

Voor alsnog laat ik belangrijke delen -zoals het inelzen van een toets- van het programma achterwege. Later zal duidelijk worden waarom dit gebeurt. Nu wil ik de aandacht nog even houden bij de controle structuren.

We zagen al eerder dat we een stukje programma kunnen schrijven dat we steeds op nieuw gebruiken. Zo kunnen we denken aan een routine die bijvoorbeeld een bepaalde tijd wacht. Als we eenmaal zo'n programma onderdeel eenmaal is geschreven, kunnen we dit gebruiken door er naar toe te springen. Een dergelijk stukje programma heet een 'routine'. Maakt hij onderdeel uit van een groter programma, dan heet het ook wel een 'subroutine'. In het hoofdprogramma kunnen we dan deze routine aanroepen door de instructie 'Jump to SubRoutine'

JSR wacht_loop in te voegen. Het programma springt dan naar de routine die 'wacht_loop' heet, en nadat deze subroutine is afgewerkt, keert het programma weer terug naar

HOOFDSTUK ZEVEN cursus MACHINETAAL.

-LOOPS-

door ard jonker

de opdracht die direct achter de JSR instructie staat.

Natuurlijk kunnen we op een bepaald moment ook wensen dat het programma ergens anders verder gaat. Dan willen we een sprong uitvoeren naar dat betreffende programma gedeelte, en niet meer terug keren naar waar we nu bezig zijn. In dat geval kunnen we de instructie JMP gebruiken; JuMP. Het programma gaat dan verder op de aangegeven plaats.

Net zoals we bij een MOVE opdracht verschillende adresseer mogelijkheden hebben gezien, kunnen we de

JMP uitvoeren op verschillende wijzen. Zo kunnen we direct springen naar een geheugenplaats (die natuurlijk een programmastukje moet bevatten), we kunnen springen naar een adres dat wordt aangegeven door de inhoud van een adres register, en we kunnen hier ook de indexed en indexed met offset gebruiken. Wat precies mogelijk is staat uitgelegd in de diverse handboeken over de 68000.

